1. Create a schema based on the given dataset

(

sl\_no int,

agent string,

date string,

login\_time string,

logout\_time string,

duration string

)

row format delimited

fields terminated by ','

stored as textfile

tblproperties("skip.header.line.count"="1")

;

2. Dump the data inside the hdfs in the given schema location.

load data local inpath '/tmp/AgentLogingReport.csv' into table agentloging\_staging;

create table agentlogin as select sl\_no, agent, from\_unixtime(unix\_timestamp(date ,'dd-MMM-yy'), 'dd-MM-yyyy') date,

from\_unixtime(unix\_timestamp(login\_time ,'HH:mm:ss'), 'HH:mm:ss') login\_time,

from\_unixtime(unix\_timestamp(logout\_time ,'HH:mm:ss'), 'HH:mm:ss') logout\_time,

from\_unixtime(unix\_timestamp(duration ,'HH:mm:ss'), 'HH:mm:ss') duration

from agentloging\_staging;

Graphical user interface

Description automatically generated

create table agentperformance\_staging

(

sl\_no int,

date string,

agent\_name string,

total\_chats int,

avg\_response\_time string,

avg\_resolution\_time string,

avg\_rating float,

total\_feedback int

)

row format delimited

fields terminated by ','

stored as textfile

tblproperties("skip.header.line.count"="1")

;

load data local inpath '/tmp/AgentPerformance.csv' into table agentperformance\_staging;

create table agentperformance as select sl\_no, from\_unixtime(unix\_timestamp(date ,'mm/dd/yyyy'), 'dd-mm-yyyy') date, agent\_name, total\_chats,

from\_unixtime(unix\_timestamp(avg\_response\_time ,'HH:mm:ss'), 'HH:mm:ss') avg\_response\_time,

from\_unixtime(unix\_timestamp(avg\_resolution\_time ,'HH:mm:ss'), 'HH:mm:ss') avg\_resolution\_time,

avg\_rating , total\_feedback

from agentperformance\_staging

;

A picture containing graphical user interface

Description automatically generated

--List of all agents' names.

select distinct agent\_name from agentperformance;

Text

Description automatically generated

--Find out agent average rating.

select agent\_name, round(avg(avg\_rating),2) average\_rating from agentperformance group by agent\_name;

Text

Description automatically generated

--Total working days for each agents

select agent, count(\*) working\_days from agentlogin group by agent;

Text

Description automatically generated

--Total query that each agent have taken

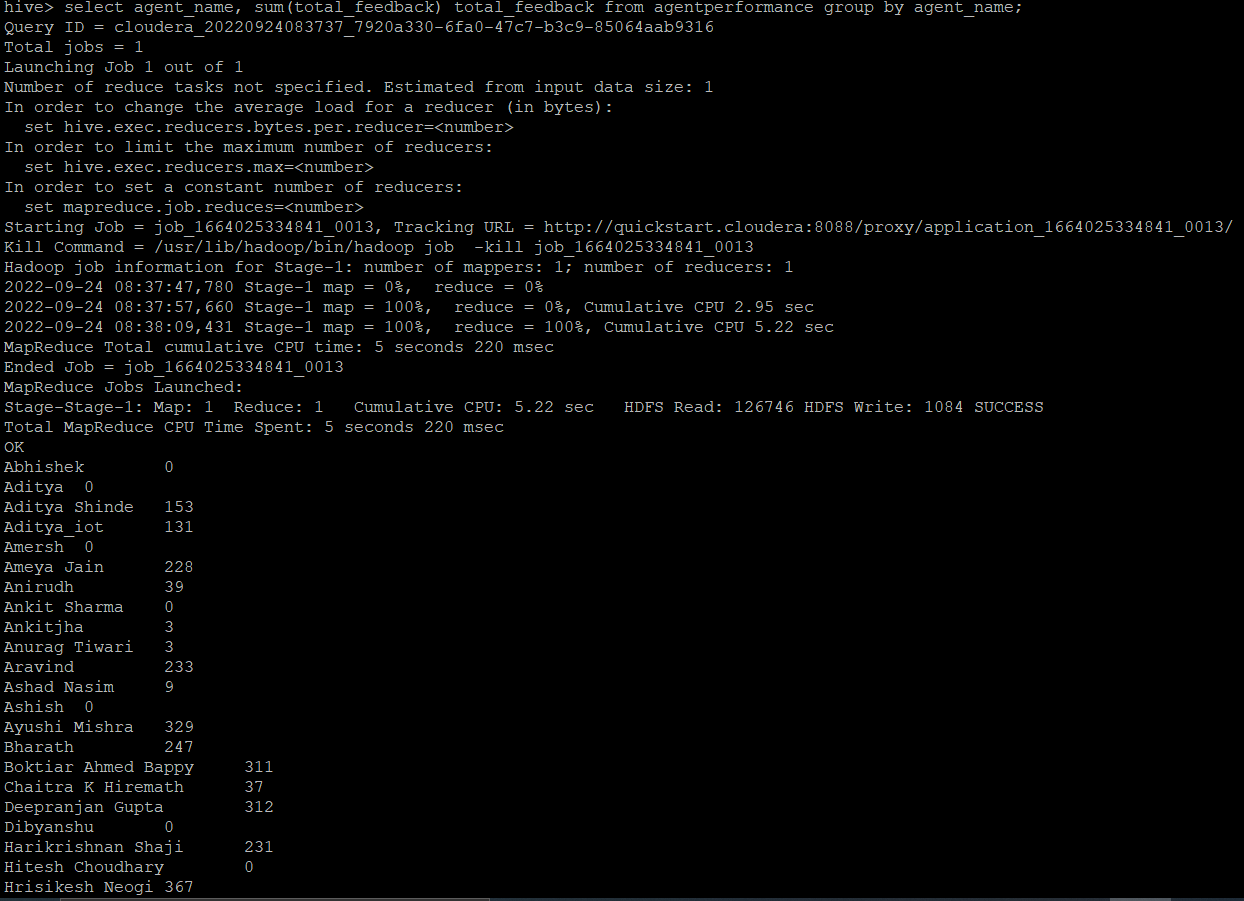
select agent\_name, sum(total\_chats) total\_chats from agentperformance group by agent\_name;

Text

Description automatically generated

--Total Feedback that each agent have received

select agent\_name, sum(total\_feedback) total\_feedback from agentperformance group by agent\_name;



--Agent name who have average rating between 3.5 to 4

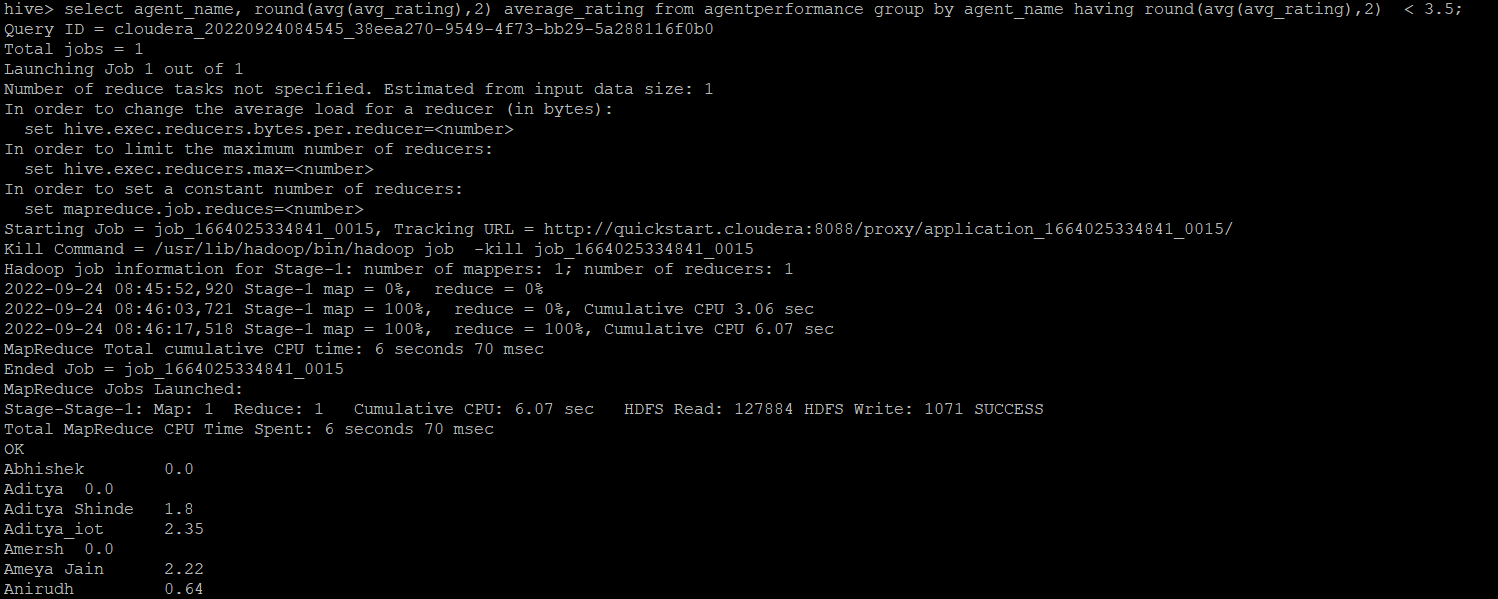
select agent\_name, avg\_rating from agentperformance where avg\_rating >= 3.5 and avg\_rating <=4;

Text

Description automatically generated

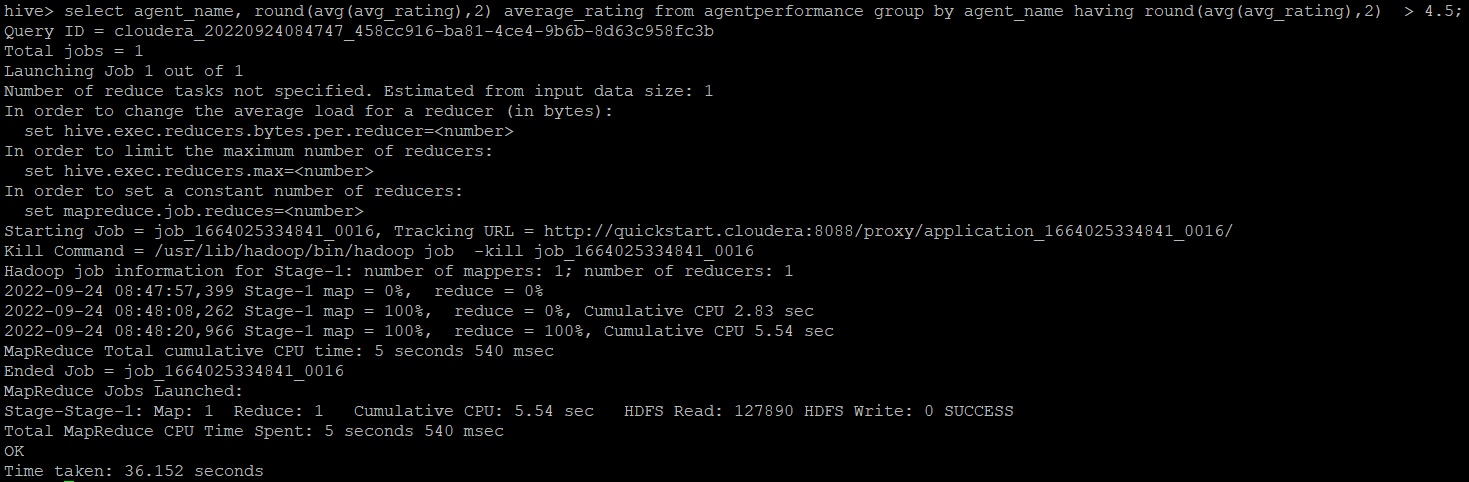
--Agent name who have rating less than 3.5

select agent\_name, round(avg(avg\_rating),2) average\_rating from agentperformance group by agent\_name having round(avg(avg\_rating),2) < 3.5;



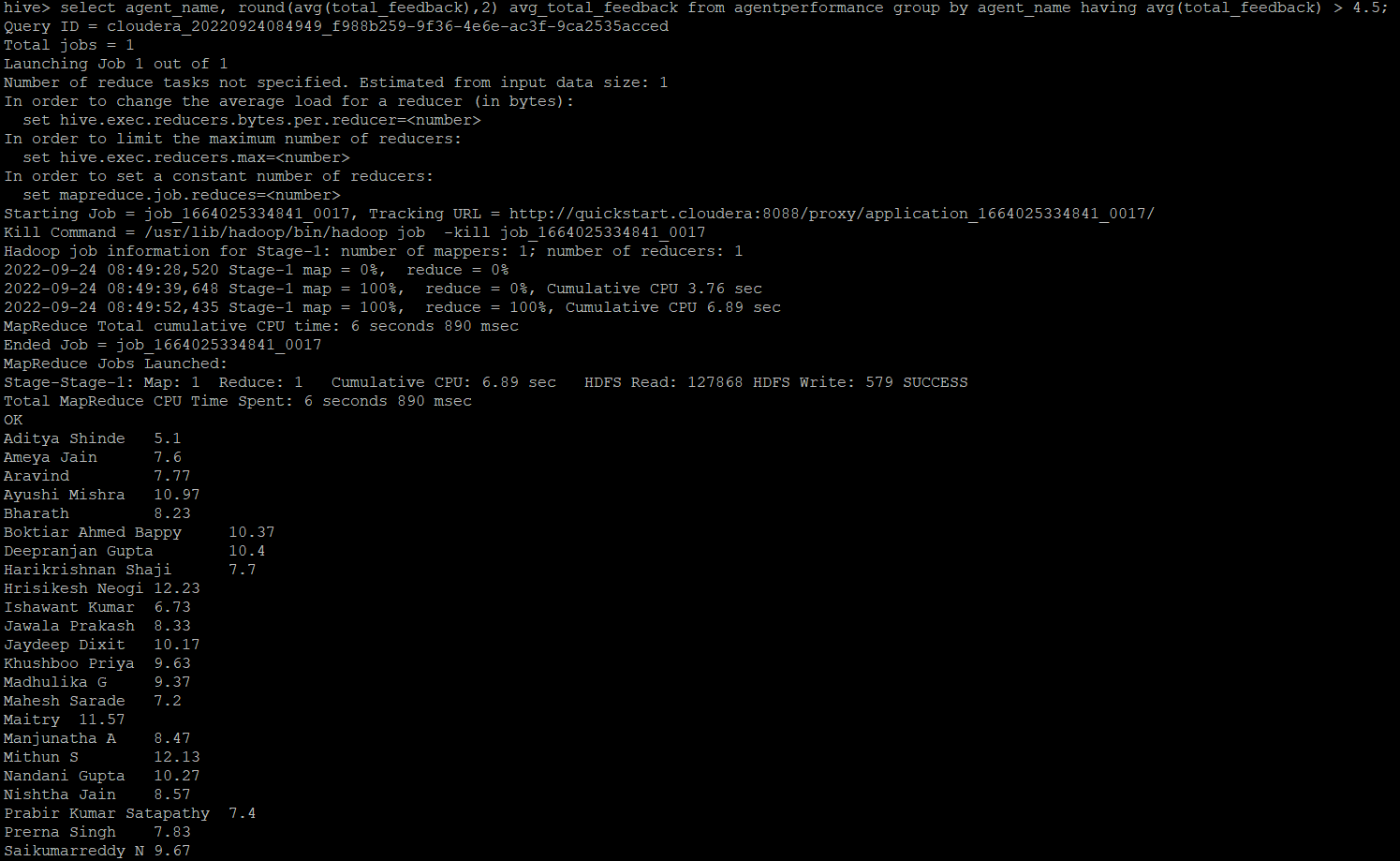
--Agent name who have rating more than 4.5

select agent\_name, round(avg(avg\_rating),2) average\_rating from agentperformance group by agent\_name having round(avg(avg\_rating),2) > 4.5;



--How many feedback agents have received more than 4.5 average

select agent\_name, round(avg(total\_feedback),2) avg\_total\_feedback from agentperformance group by agent\_name having avg(total\_feedback) > 4.5;



--average weekly response time for each agent

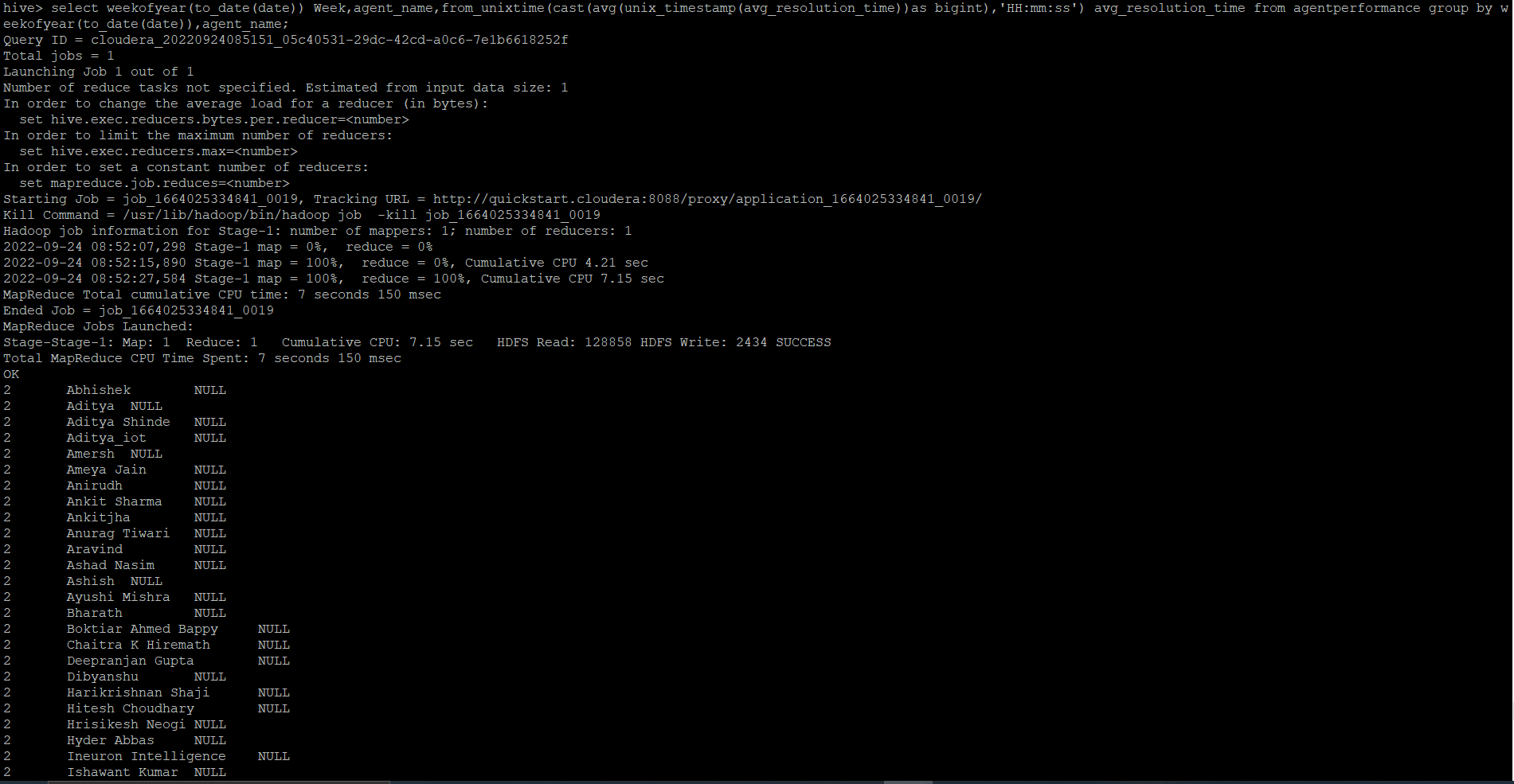
select weekofyear(to\_date(date)) Week,agent\_name,from\_unixtime(cast(avg(unix\_timestamp(avg\_response\_time,'HH:mm:ss'))as bigint),'HH:mm:ss') avg\_response\_time from agentperformance group by weekofyear(to\_date(date)),agent\_name;

Text

Description automatically generated

--average weekly resolution time for each agents

select weekofyear(to\_date(date)) Week,agent\_name,from\_unixtime(cast(avg(unix\_timestamp(avg\_resolution\_time))as bigint),'HH:mm:ss') avg\_resolution\_time from agentperformance group by weekofyear(to\_date(date)),agent\_name;



--Find the number of chat on which they have received a feedback

select total\_feedback,count(total\_chats) no\_of\_chats from agentperformance where total\_feedback !=0 group by total\_feedback;

Text

Description automatically generated

--Perform partitioning on top of the agent column and then on top of that perform bucketing for each partitioning.

set hive.exec.dynamic.partition.mode = nonstrict;

set hive.enforce.bucketing = true;

create table agentperformance\_part

(

sl\_no int,

date string,

total\_chats int,

avg\_response\_time string,

avg\_resolution\_time string,

avg\_rating float,

total\_feedback int

)

partitioned by (agent\_name string)

clustered by (date)

sorted by (date)

into 5 buckets

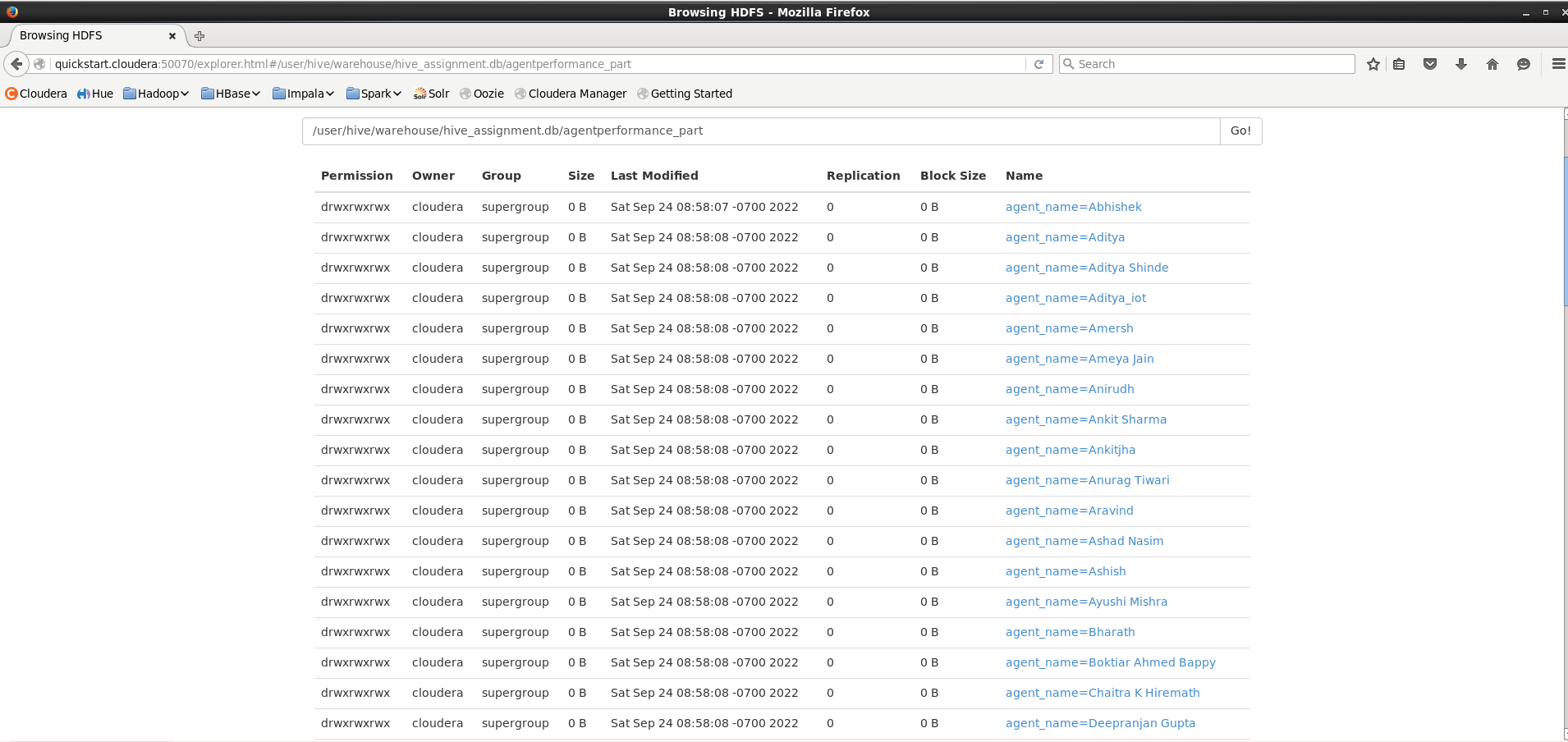
;

insert overwrite table agentperformance\_part

partition(agent\_name)

select sl\_no, date, total\_chats, avg\_response\_time, avg\_resolution\_time, avg\_rating, total\_feedback, agent\_name

from agentperformance;



Table

Description automatically generated